A comparison of 32bx32b Dadda multiplier and “pre-sum before Dadda tree” multiplier.

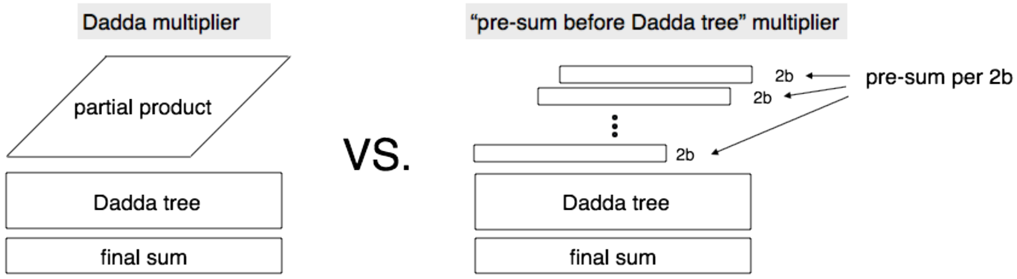
Hiro Mori
Oct. 28,2022
gmail:bqe10133@gmail.com
twitter:ubukuproject

SUMMARY

“pre-sum before Dadda tree” multiplier is 20% smaller than Dadda multiplier. The delays are the same.

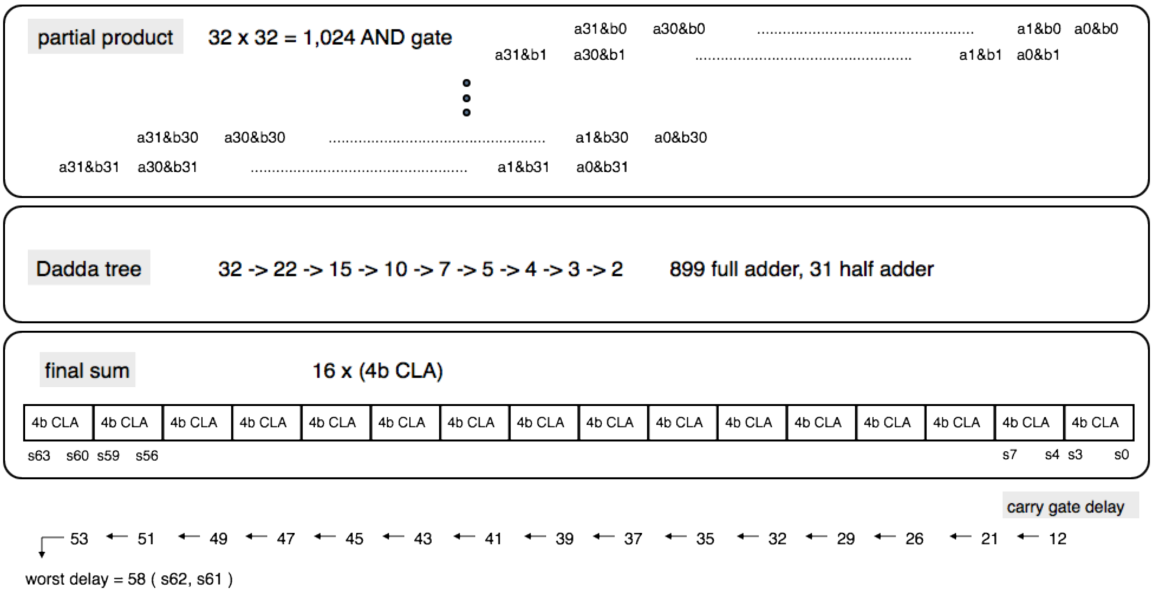
gate count		
Dadda	pre-sum before Dadda tree	percentage %
11,098	8,929	80.5%

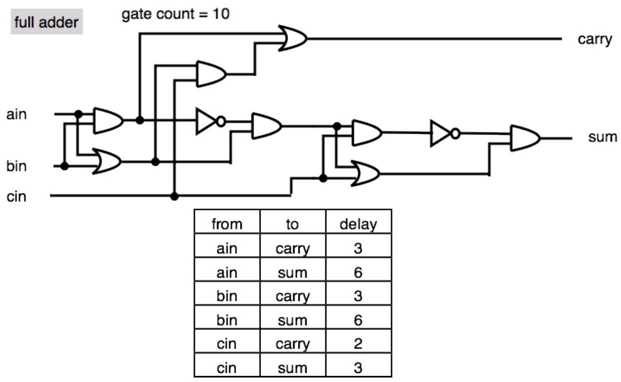
gate delay		
Dadda	pre-sum before Dadda tree	percentage %
58	58	100.0%



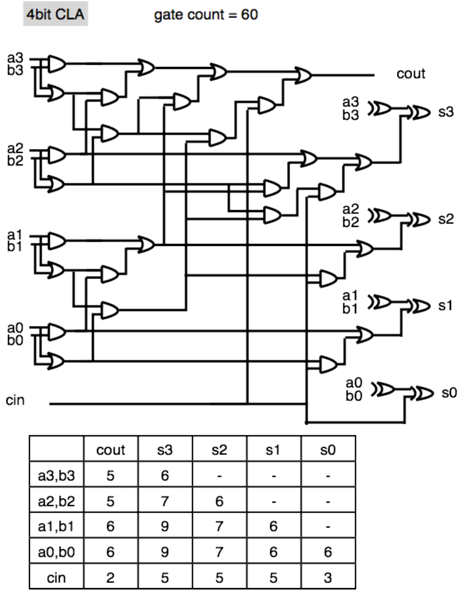
1. Dadda multiplier

1.1 multiplier diagram

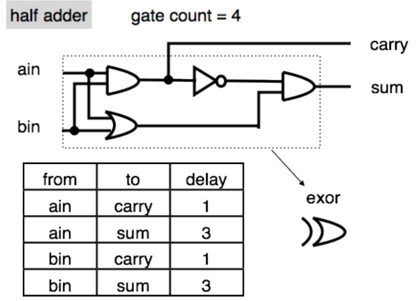




from	to	delay
ain	carry	3
ain	sum	6
bin	carry	3
bin	sum	6
cin	carry	2
cin	sum	3



	cout	s3	s2	s1	s0
a3,b3	5	6	-	-	-
a2,b2	5	7	6	-	-
a1,b1	6	9	7	6	-
a0,b0	6	9	7	6	6
cin	2	5	5	5	3



from	to	delay
ain	carry	1
ain	sum	3
bin	carry	1
bin	sum	3

1.2 gate count

[1] partial product
AND gate(= 1 gate count) x 32 x 32 = 1,024

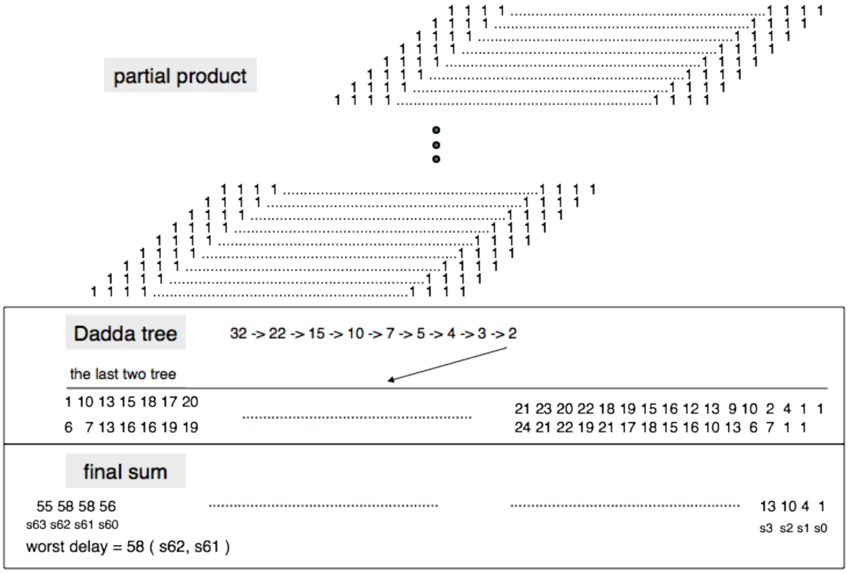
[2] Dadda tree
full adder(= 10 gate count) x 899 = 8,990
half adder(= 4 gate count) x 31 = 124

[3] final sum
4bCLA adder(= 60 gate count) x 16 = 960

[4] total gate sount
1,024 + 8,990 + 124 + 960 = **11,098**

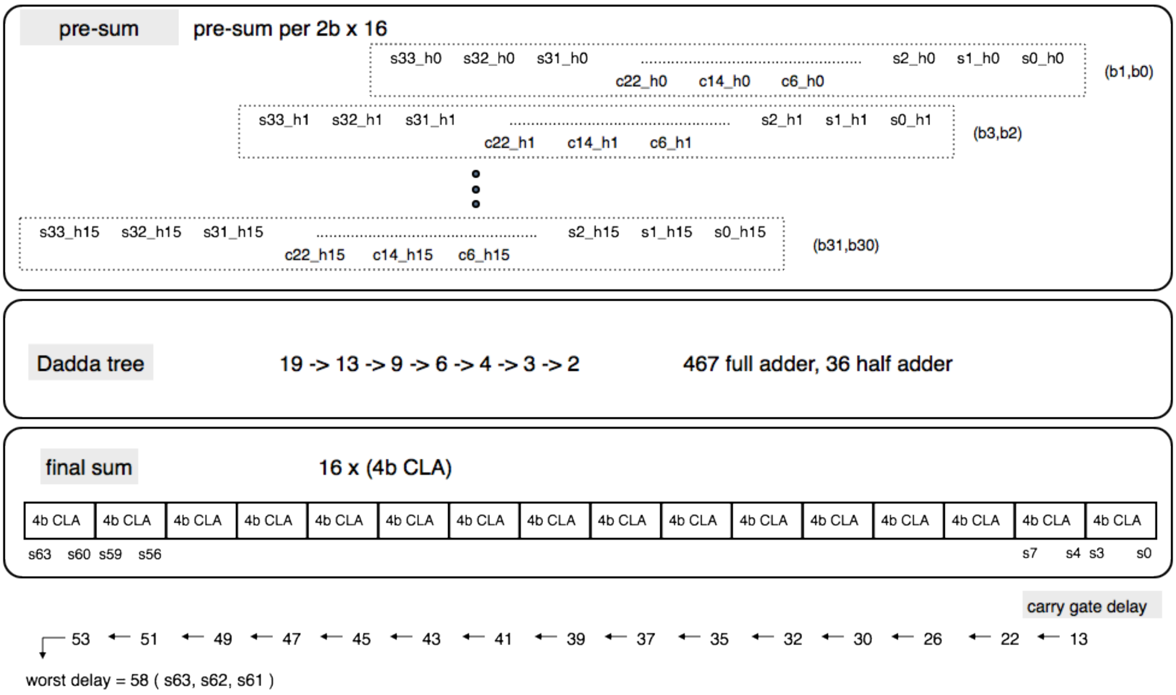
1.3 gate delay

The worst delay is 58(s62,s61).



2 “pre-sum before Dadda tree” multiplier

2.1 multiplier diagram

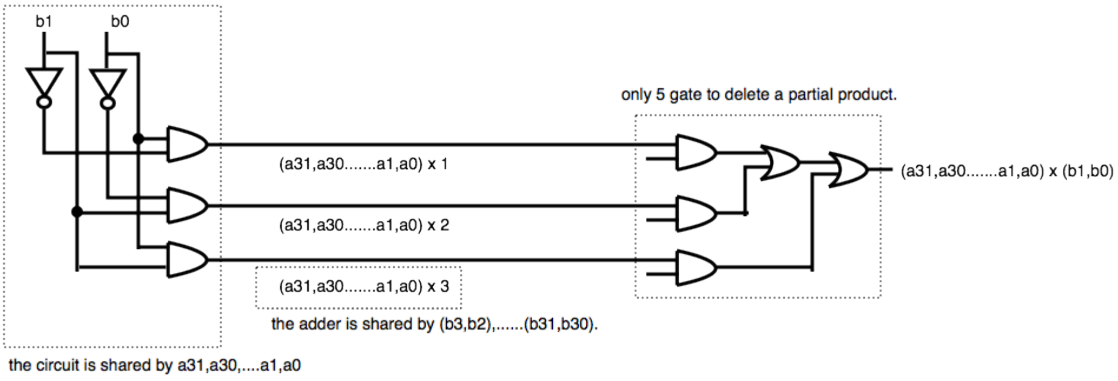


2.2 pre-sum

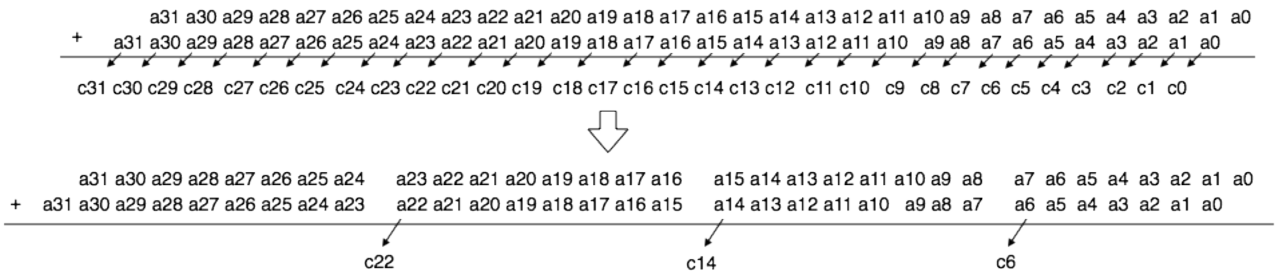
example of “(a31,a30,....a1,a0) x (b1,b0)”

b1	b0	sum
0	0	0
0	1	(a31,a30.......a1,a0) x 1
1	0	(a31,a30.......a1,a0) x 2
1	1	(a31,a30.......a1,a0) x 3

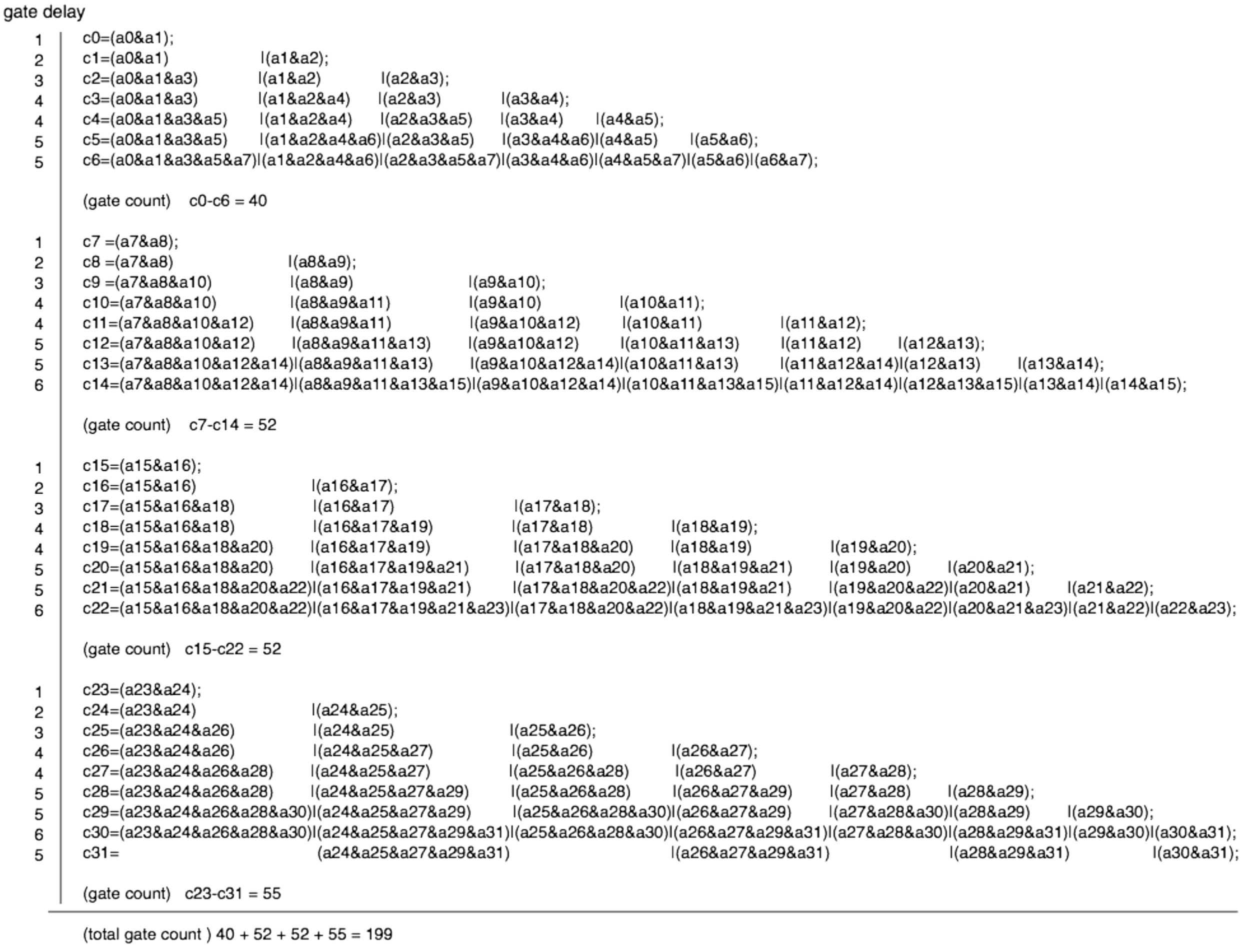
← adder


b1
b0
only 5 gate to delete a partial product.
(a31,a30.......a1,a0) x 1
(a31,a30.......a1,a0) x 2
(a31,a30.......a1,a0) x 3
(a31,a30.......a1,a0) x (b1,b0)
the adder is shared by (b3,b2),......(b31,b30).
the circuit is shared by a31,a30,....a1,a0

[1] (a31,a30,....a1,a0) x 3


a31 a30 a29 a28 a27 a26 a25 a24 a23 a22 a21 a20 a19 a18 a17 a16 a15 a14 a13 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0
+ a31 a30 a29 a28 a27 a26 a25 a24 a23 a22 a21 a20 a19 a18 a17 a16 a15 a14 a13 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0
c31 c30 c29 c28 c27 c26 c25 c24 c23 c22 c21 c20 c19 c18 c17 c16 c15 c14 c13 c12 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0
a31 a30 a29 a28 a27 a26 a25 a24 a23 a22 a21 a20 a19 a18 a17 a16 a15 a14 a13 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0
+ a31 a30 a29 a28 a27 a26 a25 a24 a23 a22 a21 a20 a19 a18 a17 a16 a15 a14 a13 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0
c22
c14
c6

The adder is cut every 8bit to reduce the carry delay. c6,c14 and c22 are added to Dadda tree.



gate delay

```
1   c0=(a0&a1);
2   c1=(a0&a1)              |(a1&a2);
3   c2=(a0&a1&a3)           |(a1&a2)          |(a2&a3);
4   c3=(a0&a1&a3)           |(a1&a2&a4)       |(a2&a3)          |(a3&a4);
4   c4=(a0&a1&a3&a5)        |(a1&a2&a4)       |(a2&a3&a5)       |(a3&a4)        |(a4&a5);
5   c5=(a0&a1&a3&a5)        |(a1&a2&a4&a6)|(a2&a3&a5)           |(a3&a4&a6)|(a4&a5)     |(a5&a6);
5   c6=(a0&a1&a3&a5&a7)|(a1&a2&a4&a6)|(a2&a3&a5&a7)|(a3&a4&a6)|(a4&a5&a7)|(a5&a6)|(a6&a7);

    (gate count)   c0-c6 = 40

1   c7 =(a7&a8);
2   c8 =(a7&a8)               |(a8&a9);
3   c9 =(a7&a8&a10)           |(a8&a9)              |(a9&a10);
4   c10=(a7&a8&a10)           |(a8&a9&a11)          |(a9&a10)          |(a10&a11);
4   c11=(a7&a8&a10&a12)       |(a8&a9&a11)          |(a9&a10&a12)      |(a10&a11)          |(a11&a12);
5   c12=(a7&a8&a10&a12)       |(a8&a9&a11&a13)      |(a9&a10&a12)      |(a10&a11&a13)      |(a11&a12)      |(a12&a13);
5   c13=(a7&a8&a10&a12&a14)|(a8&a9&a11&a13)         |(a9&a10&a12&a14)|(a10&a11&a13)        |(a11&a12&a14)|(a12&a13)      |(a13&a14);
6   c14=(a7&a8&a10&a12&a14)|(a8&a9&a11&a13&a15)|(a9&a10&a12&a14)|(a10&a11&a13&a15)|(a11&a12&a14)|(a12&a13&a15)|(a13&a14)|(a14&a15);

    (gate count)   c7-c14 = 52

1   c15=(a15&a16);
2   c16=(a15&a16)                |(a16&a17);
3   c17=(a15&a16&a18)            |(a16&a17)                |(a17&a18);
4   c18=(a15&a16&a18)            |(a16&a17&a19)            |(a17&a18)             |(a18&a19);
4   c19=(a15&a16&a18&a20)        |(a16&a17&a19)            |(a17&a18&a20)         |(a18&a19)             |(a19&a20);
5   c20=(a15&a16&a18&a20)        |(a16&a17&a19&a21)        |(a17&a18&a20)         |(a18&a19&a21)         |(a19&a20)      |(a20&a21);
5   c21=(a15&a16&a18&a20&a22)|(a16&a17&a19&a21)            |(a17&a18&a20&a22)|(a18&a19&a21)              |(a19&a20&a22)|(a20&a21)      |(a21&a22);
6   c22=(a15&a16&a18&a20&a22)|(a16&a17&a19&a21&a23)|(a17&a18&a20&a22)|(a18&a19&a21&a23)|(a19&a20&a22)|(a20&a21&a23)|(a21&a22)|(a22&a23);

    (gate count)   c15-c22 = 52

1   c23=(a23&a24);
2   c24=(a23&a24)                |(a24&a25);
3   c25=(a23&a24&a26)            |(a24&a25)                |(a25&a26);
4   c26=(a23&a24&a26)            |(a24&a25&a27)            |(a25&a26)             |(a26&a27);
4   c27=(a23&a24&a26&a28)        |(a24&a25&a27)            |(a25&a26&a28)         |(a26&a27)             |(a27&a28);
5   c28=(a23&a24&a26&a28)        |(a24&a25&a27&a29)        |(a25&a26&a28)         |(a26&a27&a29)         |(a27&a28)      |(a28&a29);
5   c29=(a23&a24&a26&a28&a30)|(a24&a25&a27&a29)            |(a25&a26&a28&a30)|(a26&a27&a29)              |(a27&a28&a30)|(a28&a29)      |(a29&a30);
6   c30=(a23&a24&a26&a28&a30)|(a24&a25&a27&a29&a31)|(a25&a26&a28&a30)|(a26&a27&a29&a31)|(a27&a28&a30)|(a28&a29&a31)|(a29&a30)|(a30&a31);
5   c31=                         (a24&a25&a27&a29&a31)                            |(a26&a27&a29&a31)                    |(a28&a29&a31)            |(a30&a31);

    (gate count)   c23-c31 = 55

    (total gate count ) 40 + 52 + 52 + 55 = 199
```

	gate delay	gate count
s0 = a0;	1	0
s1 = (a0^a1);	3	4
s2 =c0^(a1^a2);	6	8
s3 =c1^(a2^a3);	6	8
s4 =c2^(a3^a4);	6	8
s5 =c3^(a4^a5);	7	8
s6 =c4^(a5^a6);	7	8
s7 =c5^(a6^a7);	8	8
s8 = (a7^a8);	3	4
s9 = c7^(a8^a9);	6	8
s10= c8^(a9^a10);	6	8
s11= c9^(a10^a11);	6	8
s12=c10^(a11^a12);	7	8
s13=c11^(a12^a13);	7	8
s14=c12^(a13^a14);	8	8
s15=c13^(a14^a15);	8	8

s16= (a15^a16);	3	4
s17=c15^(a16^a17);	6	8
s18=c16^(a17^a18);	6	8
s19=c17^(a18^a19);	6	8
s20=c18^(a19^a20);	7	8
s21=c19^(a20^a21);	7	8
s22=c20^(a21^a22);	8	8
s23=c21^(a22^a23);	8	8
s24= (a23^a24);	3	4
s25=c23^(a24^a25);	6	8
s26=c24^(a25^a26);	6	8
s27=c25^(a26^a27);	6	8
s28=c26^(a27^a28);	7	8
s29=c27^(a28^a29);	7	8
s30=c28^(a29^a30);	8	8
s31=c29^(a30^a31);	8	8
s32=c30^a31;	9	4
s33=c31;	5	0
		(tota1 gate count) 236

The total gate count of “(a31,a30,....a1,a0) x 3” = 199 + 236 = 435

[2] (a31,a30,....a1,a0) x (b1,b0)

	gate delay	gate count
h0_11 = b1&b0;	1	1
h0_10 = b1&~b0;	2	2
h0_01 = ~b1&b0;	2	2
s0_h0 = (h0_11&s0) |(h0_01&a0);	4	3
s1_h0 = (h0_11&s1) |(h0_01&a1) |(h0_10&a0);	5	5
s2_h0 = (h0_11&s2) |(h0_01&a2) |(h0_10&a1);	8	5
s3_h0 = (h0_11&s3) |(h0_01&a3) |(h0_10&a2);	8	5
s4_h0 = (h0_11&s4) |(h0_01&a4) |(h0_10&a3);	8	5
s5_h0 = (h0_11&s5) |(h0_01&a5) |(h0_10&a4);	9	5
s6_h0 = (h0_11&s6) |(h0_01&a6) |(h0_10&a5);	9	5
s7_h0 = (h0_11&s7) |(h0_01&a7) |(h0_10&a6);	10	5
s8_h0 = (h0_11&s8) |(h0_01&a8) |(h0_10&a7);	5	5
s9_h0 = (h0_11&s9) |(h0_01&a9) |(h0_10&a8);	8	5
s10_h0 = (h0_11&s10)|(h0_01&a10)|(h0_10&a9);	8	5
s11_h0 = (h0_11&s11)|(h0_01&a11)|(h0_10&a10);	8	5
s12_h0 = (h0_11&s12)|(h0_01&a12)|(h0_10&a11);	9	5
s13_h0 = (h0_11&s13)|(h0_01&a13)|(h0_10&a12);	9	5
s14_h0 = (h0_11&s14)|(h0_01&a14)|(h0_10&a13);	10	5
s15_h0 = (h0_11&s15)|(h0_01&a15)|(h0_10&a14);	10	5
s16_h0 = (h0_11&s16)|(h0_01&a16)|(h0_10&a15);	5	5
s17_h0 = (h0_11&s17)|(h0_01&a17)|(h0_10&a16);	8	5
s18_h0 = (h0_11&s18)|(h0_01&a18)|(h0_10&a17);	8	5
s19_h0 = (h0_11&s19)|(h0_01&a19)|(h0_10&a18);	8	5
s20_h0 = (h0_11&s20)|(h0_01&a20)|(h0_10&a19);	9	5
s21_h0 = (h0_11&s21)|(h0_01&a21)|(h0_10&a20);	9	5
s22_h0 = (h0_11&s22)|(h0_01&a22)|(h0_10&a21);	10	5
s23_h0 = (h0_11&s23)|(h0_01&a23)|(h0_10&a22);	10	5
s24_h0 = (h0_11&s24)|(h0_01&a24)|(h0_10&a23);	5	5
s25_h0 = (h0_11&s25)|(h0_01&a25)|(h0_10&a24);	8	5
s26_h0 = (h0_11&s26)|(h0_01&a26)|(h0_10&a25);	8	5
s27_h0 = (h0_11&s27)|(h0_01&a27)|(h0_10&a26);	8	5
s28_h0 = (h0_11&s28)|(h0_01&a28)|(h0_10&a27);	9	5
s29_h0 = (h0_11&s29)|(h0_01&a29)|(h0_10&a28);	9	5
s30_h0 = (h0_11&s30)|(h0_01&a30)|(h0_10&a29);	10	5
s31_h0 = (h0_11&s31)|(h0_01&a31)|(h0_10&a30);	10	5
s32_h0 = (h0_11&s32) |(h0_10&a31);	11	3
s33_h0 = (h0_11&s33);	6	1
c6_h0 = c6&h0_11;	6	1
c14_h0 = c14&h0_11;	7	1
c22_h0 = c22&h0_11;	7	1
		(total gate count) 170

The total gate count of “(a31,a30,....a1,a0) x (b1,b0)” = 170

2.3 gate count

[1] pre-sum
(a31,a30,..a1,a0) x 3 = 236 + 199 = 435
pre-sum per 2b x 16 = 170 x 16 = 2,720

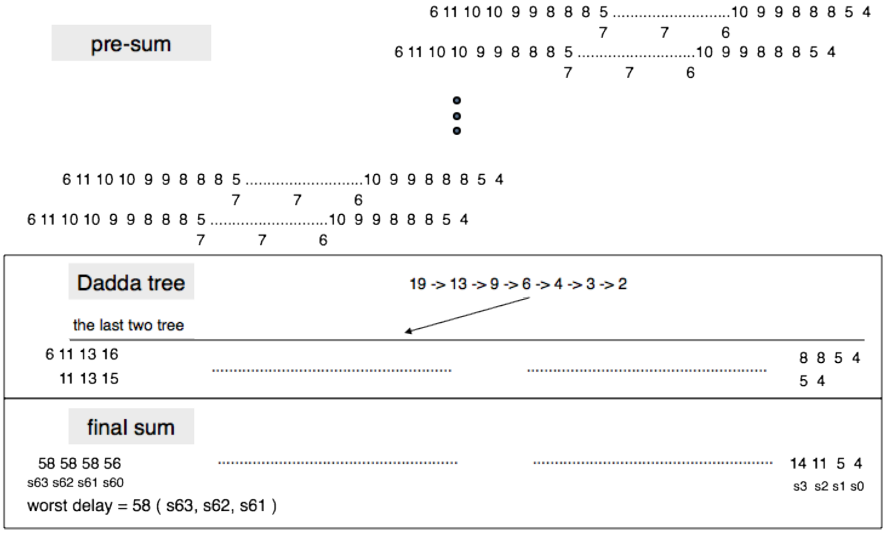
[2] Dadda tree
full adder(= 10 gate count) x 467 = 4,670
half adder(= 4 gate count) x 36 = 144

[3] final sum
4bCLA adder(= 60 gate count) x 16 = 960

[4] total gate count
435 + 2,720 + 4,670 + 144 + 960 = **8,929**

2.4 gate delay

The worst delay is 58(s63,s62,s61).



3 Dadda tree calculation

[1] Dadda tree reduction program

[example]

9 input   3 full adder

```
input.txt
15 11 12 15 13 12 12 13 12 Z
```

```
--------------------------------------------
%./daddatree  input.txt  3

bufinlen:9
15 11 12 15 13 12 12 13 12
bufinlen:9
15 15 13 13 12 12 12 12 11
bufinlen:9
15000 15000 13000 13000 12000 12000 12000 12000 11000
first_val:15 second_val:13,third_val:13
carry:17 sum:19
first_val:15 second_val:12,third_val:12
carry:17 sum:18
first_val:12 second_val:12,third_val:11
carry:15 sum:18
```

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <string>
#include <math.h>
using namespace std;
void daddasort(int bufinlen,int *bufin);
void daddafa(int first_val,int second_val,int third_val);
void daddatreecalc(int *mainbuf,int fanum);

int main(int argc, char * const argv[]) {

  int  mainbuf[1024];//equation main
        char c0;
        int  d0;
        int  i;
        int  fanum;

        ifstream fv0(argv[1]);//equation

        for(i=0;i<1024;i++){
                mainbuf[i]=0;
        }
        for(i=0;i<1024;i++){
                fv0.get(c0);
                d0=(int)c0;
                mainbuf[i]=d0;
                if(d0==90){break;}
        }

        fanum = atoi(argv[2]);
        daddatreecalc(mainbuf,fanum);

        fv0.close();
        return 0;
}

//--------------------------------
//function : daddatreecalculation
// 3bit x n -> 2bit x n
//--------------------------------
void daddasort(int bufinlen,int *bufin){

        int i,j,k;
        int min,tmp;
        int bufintmp[128];

        for(i=0;i<bufinlen-1;i++){
                min=bufin[i];
                k=i;
                for(j=i+1;j<bufinlen;j++){
                        if(bufin[j]<min){
                                min=bufin[j];
                                k=j;
                        }
                }
                tmp=bufin[i];
                bufin[i]=bufin[k];
                bufin[k]=tmp;
        }

        //sort reverse
        for(i=0;i<bufinlen;i++){
                bufintmp[bufinlen-1-i]=bufin[i];
        }
        for(i=0;i<bufinlen;i++){
                bufin[i]=bufintmp[i];
        }

}

void daddafa(int first_val,int second_val,int third_val){

        int first_carry,first_sum;
        int second_carry,second_sum;

        first_carry=first_val+2;
        first_sum=first_val+3;
        second_carry=second_val+3;
        second_sum=second_val+6;

        if(first_carry>=second_carry){
                printf("carry:%d ",first_carry);
        }
        else{
                printf("carry:%d ",second_carry);
        }

        if(first_sum>=second_sum){
                printf("sum:%d\n",first_sum);
        }
        else{
                printf("sum:%d\n",second_sum);
        }
}


void daddatreecalc(int *mainbuf,int fanum){

        int i,j,m;
        int bufin[128];
        int bufinlen;
        int flag;
        int first_val,second_val,third_val;
        int daddavalue;

        //-----------------------------
        // store the value in bufin[].
        //-----------------------------
        m=0;
        daddavalue=0;
        for(i=0;i<32768;i++){
                if(mainbuf[i]==90){ // char "Z" = 90
                        break;
                }
                else if(mainbuf[i]==32){ // char " " = 32
                        bufin[m]=daddavalue;
                        m++;
                        daddavalue=0;
                }
                else{
                        daddavalue=daddavalue*10;
                        daddavalue=daddavalue+(mainbuf[i]-48);//char "0" = 48
                }
        }
        bufinlen=m;

        //-----------------------------
        // calculation
        //-----------------------------
        printf("bufinlen:%d\n",bufinlen);
        for(i=0; i<bufinlen; i++) {
                printf("%d ",bufin[i]);
        }
        printf("\n");

        daddasort(bufinlen, bufin);

        printf("bufinlen:%d\n",bufinlen);
        for(i=0; i<bufinlen; i++) {
                printf("%d ",bufin[i]);
        }
        printf("\n");

        flag=0;

        for(i=0;i<bufinlen;i++){
                if(i>=(bufinlen-fanum*3)){
                        bufin[i]=bufin[i]*1000;
                }
                else{
                        bufin[i]=bufin[i];
                }
        }

        printf("bufinlen:%d\n",bufinlen);
        for(i=0; i<bufinlen; i++) {
                printf("%d ",bufin[i]);
        }
        printf("\n");

        for(i=0;i<fanum;i++){
                daddasort(bufinlen, bufin);
                flag=0;

                for(j=0;j<bufinlen;j++){
                        if(j==0){
                                bufin[j]=bufin[j]/1000;
                                first_val=bufin[j];
                        }
                        else if(j>0 & j<(fanum-i)){
                                if((bufin[j]/1000<=(bufin[0]-3)) & flag==0){
                                        flag=1;
                                        bufin[j]=bufin[j]/1000;
                                        bufin[j+1]=bufin[j+1]/1000;
                                        second_val=bufin[j];
                                        third_val=bufin[j+1];
                                }
                        }
                        else if(j==(fanum-i)){
                                if(flag==0){
                                        if(bufin[j]==bufin[j-1]){
                                                flag=1;
                                                bufin[j]=bufin[j]/1000;
                                                bufin[j-1]=bufin[j-1]/1000;
                                                second_val=bufin[j];
                                                third_val=bufin[j-1];
                                        }
                                        else{
                                                flag=1;
                                                bufin[j]=bufin[j]/1000;
                                                bufin[j+1]=bufin[j+1]/1000;
                                                second_val=bufin[j];
                                                third_val=bufin[j+1];
                                        }
                                }
                        }
                }
                printf("first_val:%d second_val:%d,third_val:%d\n",first_val,second_val,third_val);
                daddafa(first_val, second_val, third_val);
        }
}
```

[2] Dadda multiplier

```
1-------------------------------   1---------------------    1 -- -- -- -- -- -- -- -- -- -- -- -- -- --
11------------------------------   11--------------------    1  1 -- -- -- -- -- -- -- -- -- -- -- -- --
111-----------------------------   111-------------------    1  1  1 -- -- -- -- -- -- -- -- -- -- -- --
1111----------------------------   1111------------------    1  1  1  1 -- -- -- -- -- -- -- -- -- -- --
11111---------------------------   11111-----------------    1  1  1  1  1 -- -- -- -- -- -- -- -- -- --
111111--------------------------   111111----------------    1  1  1  1  1  1 -- -- -- -- -- -- -- -- --
1111111-------------------------   1111111---------------    1  1  1  1  1  1  1 -- -- -- -- -- -- -- --
11111111------------------------   11111111--------------    1  1  1  1  1  1  1  1 -- -- -- -- -- -- --
111111111-----------------------   111111111-------------    1  1  1  1  1  1  1  1  1 -- -- -- -- -- --
1111111111----------------------   1111111111------------    1  1  1  1  1  1  1  1  1  1 -- -- -- -- --
11111111111---------------------   11111111111-----------    1  1  1  1  1  1  1  1  1  1  1 -- -- -- --
111111111111--------------------   111111111111----------    1  1  1  1  1  1  1  1  1  1  1  1 -- -- --
1111111111111-------------------   1111111111111---------    1  1  1  1  1  1  1  1  1  1  1  1  1 -- --
11111111111111------------------   11111111111111--------    1  1  1  1  1  1  1  1  1  1  1  1  1  1 --
111111111111111-----------------   111111111111111-------    1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1111111111111111----------------   1111111111111111------    4  1  1  1  1  1  1  1  1  1  1  1  1  1  1
11111111111111111---------------   11111111111111111-----    2  4  7  1  1  1  1  1  1  1  1  1  1  1  1
111111111111111111--------------   111111111111111111----    4  2  4  7  7  1  1  1  1  1  1  1  1  1  1
1111111111111111111-------------   1111111111111111111---    2  4  7  4  4  7  7  1  1  1  1  1  1  1  1
11111111111111111111------------   11111111111111111111--    4  2  4  7  7  4  4  7  7  1  1  1  1  1  1
111111111111111111111-----------   111111111111111111111-    2  4  7  4  4  7  7  4  4  7  7  1  1  1  1
1111111111111111111111----------   1111111111111111111111    4  2  4  7  7  4  4  7  7  4  4  7  7  1  1
11111111111111111111111---------   4111111111111111111111    2  7  7  4  4  7  7  4  4  7  7  4  4  7  4
111111111111111111111111--------   2471111111111111111111    7  4  4  7  7  4  4  7  7  4  4  7  7  4  7
1111111111111111111111111-------   4247711111111111111111    6 10  7  4  4  7  7  4  4  7  7  4  4  7  7
11111111111111111111111111------   2474477111111111111111   10  9  4 10  7  4  4  7  7  4  4  7  7  4  7
111111111111111111111111111-----   4247744771111111111111    9 10 10  9  4 10  7  4  4  7  7  4  4  7  7
1111111111111111111111111111----   2474477447711111111111   10  9  9 10 10  9  4 10  7  4  4  7  7  4  7
11111111111111111111111111111---   4247744774477111111111    9 10 10  9  9 10 10  9  4 10  7  4  4  7  7
111111111111111111111111111111--   2474477447744771111111   10  9  9 10 10  9  9 10 10  9  4 10  7  4  7
1111111111111111111111111111111-   4247744774477447711111    9 10 10  9  9 10 10  9  9 10 10  9  4 10  7
11111111111111111111111111111111   2474477447744774477111   13  9  9 10 10  9  9 10 10  9  9 10  7  9  7
-1111111111111111111111111111111   4247744774477447744711   10 13 10  9  9 10 10  9  9 10 10  9  4  7  7
--111111111111111111111111111111   2774477447744774471411   13 10  9 10 10  9  9 10 10  9  9 10  7  4  7
---11111111111111111111111111111   7447744774477447141111   10 10 10  9  9 10 10  9  9 10 10  9  4 10  7
----1111111111111111111111111111   4774477447744714111111   10  9  9 10 10  9  9 10 10  9  9 10  7  9  7
-----111111111111111111111111111   7447744774471411111111    9 10 10  9  9 10 10  9  9 10  7  9  4  7  7
------11111111111111111111111111   4774477447141111111111   10  9  9 10 10  9  9 10  7  9  4  7  7  4  7
-------1111111111111111111111111   7447744714111111111111    9 10 10  9  9 10  7  9  4  7  7  4  4  7  7
--------111111111111111111111111   4774471411111111111111   10  9  9 10  7  9  4  7  7  4  4  7  7  4  7
---------11111111111111111111111   7447141111111111111111    9 10  7  9  4  7  7  4  4  7  7  4  4  7  7
----------1111111111111111111111   4714111111111111111111    7  9  6  7  7  4  4  7  7  4  4  7  7  4  7
-----------111111111111111111111   1411111111111111111111    6  7  7  6  4  7  7  4  4  7  7  4  4  7  4
------------11111111111111111111   11111111111111111111--    7  4  4  7  7  4  4  7  7  4  4  7  1  4  1
-------------1111111111111111111   1111111111111111111---    4  7  7  4  4  7  7  4  4  7  1  4  1  1  1
--------------111111111111111111   111111111111111111----    7  4  4  7  7  4  4  7  1  4  1  1  1  1  1
---------------11111111111111111   11111111111111111-----    4  7  7  4  4  7  1  4  1  1  1  1  1  1  1
----------------1111111111111111   1111111111111111------    7  4  4  7  1  4  1  1  1  1  1  1  1  1  1
-----------------111111111111111   111111111111111-------    4  7  1  4  1  1  1  1  1  1  1  1  1  1  1
------------------11111111111111   11111111111111--------    1  4  1  1  1  1  1  1  1  1  1  1  1  1  1
-------------------1111111111111   1111111111111---------    1  1  1  1  1  1  1  1  1  1  1  1  1 -- --
--------------------111111111111   111111111111----------    1  1  1  1  1  1  1  1  1  1  1  1 -- -- --
---------------------11111111111   11111111111-----------    1  1  1  1  1  1  1  1  1  1  1 -- -- -- --
----------------------1111111111   1111111111------------    1  1  1  1  1  1  1  1  1  1 -- -- -- -- --
-----------------------111111111   111111111-------------    1  1  1  1  1  1  1  1  1 -- -- -- -- -- --
------------------------11111111   11111111--------------    1  1  1  1  1  1  1  1 -- -- -- -- -- -- --
-------------------------1111111   1111111---------------    1  1  1  1  1  1  1 -- -- -- -- -- -- -- --
--------------------------111111   111111----------------    1  1  1  1  1  1 -- -- -- -- -- -- -- -- --
---------------------------11111   11111-----------------    1  1  1  1  1 -- -- -- -- -- -- -- -- -- --
----------------------------1111   1111------------------    1  1  1  1 -- -- -- -- -- -- -- -- -- -- --
-----------------------------111   111-------------------    1  1  1 -- -- -- -- -- -- -- -- -- -- -- --
------------------------------11   11--------------------    1  1 -- -- -- -- -- -- -- -- -- -- -- -- --
-------------------------------1   1---------------------    1 -- -- -- -- -- -- -- -- -- -- -- -- -- --
--------------------------------   ----------------------    1 -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
 1 -- -- -- -- -- -- -- -- --    1 -- -- -- -- -- --    1 -- -- -- --    1 -- -- --    1 -- --    1 --    1
 1  1 -- -- -- -- -- -- -- --    1  1 -- -- -- -- --    1  1 -- -- --    1  1 -- --    1  1 --    1  1    4
 1  1  1 -- -- -- -- -- -- --    1  1  1 -- -- -- --    1  1  1 -- --    1  1  1 --    1  1  1    4  1   10
 1  1  1  1 -- -- -- -- -- --    1  1  1  1 -- -- --    1  1  1  1 --    1  1  1  1    4  1  1    2  7   13
 1  1  1  1  1 -- -- -- -- --    1  1  1  1  1 -- --    1  1  1  1  1    4  1  1  1    2  7  4   10  6   16
 1  1  1  1  1  1 -- -- -- --    1  1  1  1  1  1 --    4  1  1  1  1    2  7  1  4    8  4  7    9 13   19
 1  1  1  1  1  1  1 -- -- --    1  1  1  1  1  1  1    2  4  7  1  1    7  4  4  7    6 10  7   13 10   20
 1  1  1  1  1  1  1  1 -- --    4  1  1  1  1  1  1    7  2  4  7  4    4 10  7  7   13  9 10   12 16   22
 1  1  1  1  1  1  1  1  1 --    2  4  7  1  1  1  1    4  7  7  4  7   10  7  7  7   10 13 10   16 15   24
 1  1  1  1  1  1  1  1  1  1    4  2  4  7  7  1  1   10  6  4  7  7    9 12  7 10   15 10 12   15 18   26
 4  1  1  1  1  1  1  1  1  1    2  7  7  4  4  7  4    9 10 10  4  7   13  9 10 10   12 16 13   19 17   26
 2  4  7  1  1  1  1  1  1  1    7  4  4  7  7  4  7   13  9  9 10  7   11 15 10 13   17 13 15   18 21   27
 4  2  4  7  7  1  1  1  1  1    6 10  7  4  4  7  7   10 13 10  7 10   16 12 10 13   15 18 16   22 19   29
 2  4  7  4  4  7  7  1  1  1   10  9  4 10  7  4  7   13 10  8 12 10   13 16 12 13   19 15 16   20 22   31
 4  2  4  7  7  4  4  7  7  1    9 10 10  9  4 10  7   12 15 13 11 10   17 13 13 15   16 19 17   23 21   31
 2  7  7  4  4  7  7  4  4  7   13  9  9 10 10  9  7   15 12 12 15 13   14 18 15 15   21 17 18   21 24   31
10  6  4  7  7  4  4  7  7  4   10 13 10  9  7 10 10   12 16 15 12 13   18 15 15 16   18 21 18   24 23   32
 9 10 10  4  4  7  7  4  4  7   13 10  9 12 10  8 10   16 13 12 15 13   15 19 15 16   21 18 19   23 25   34
10  9  9 10 10  4  4  7  7  4   12 15 13 11  9 10 10   14 17 16 12 15   20 16 16 17   18 22 20   26 23   34
 9 10 10  9  9 10 10  4  4  7   15 12 12 15 13  7 10   18 15 14 16 15   17 21 16 18   23 19 21   24 27   34
10  9  9 10 10  9  9 10 10  4   12 16 15 12 12 15 10   17 18 18 15 16   22 18 18 18   20 24 22   28 25   35
 9 13 10  9  9 10 10  9  9  7   15 13 12 15 15 12 13   19 17 17 18 15   19 23 18 19   25 21 23   26 29   37
13 10  9 10 10  9  9 10 10  4   12 16 15 12 12 15 13   17 19 18 17 16   23 20 18 19   22 25 23   29 27   37
10 13 13  9  9 10 10  9  9 10   16 13 12 15 15 12 13   19 17 17 18 16   20 23 18 19   25 22 23   27 29   38
```

13 10 10 13 10 9 9 10 10 7	15 16 16 12 12 15 13	17 21 18 17 16	23 20 18 21	22 26 23	29 27	38
12 13 13 10 9 12 10 9 8 10	16 15 13 16 15 12 13	21 18 15 19 16	20 22 19 21	26 23 22	28 29	40
13 12 12 13 13 11 9 12 12 9	15 18 18 14 14 15 13	18 21 20 16 18	24 20 20 21	23 26 24	30 28	40
12 15 13 12 12 13 13 11 11 12	18 15 15 18 17 13 15	21 20 17 21 18	21 24 21 21	27 23 24	28 30	40
15 12 12 15 13 12 12 13 13 11	15 19 18 15 15 18 15	20 21 21 19 19	25 22 21 21	24 27 25	31 29	40
12 16 15 12 12 15 13 12 12 12	18 17 15 18 18 15 16	23 20 20 21 18	22 26 21 23	28 24 26	29 32	42
16 13 12 15 15 12 12 15 13 11	17 19 18 17 15 18 16	20 23 22 20 19	26 23 22 23	25 29 26	32 30	42
13 16 15 12 12 15 15 12 12 13	19 17 17 18 18 17 16	23 20 20 23 19	23 26 23 23	29 26 26	31 32	42
16 15 12 15 15 12 12 15 13 11	17 21 18 17 17 18 16	20 23 23 20 21	26 23 23 23	26 29 26	32 31	42
15 16 15 12 12 15 15 12 12 13	21 18 17 18 18 15 16	24 20 20 22 21	23 26 22 24	29 26 26	31 32	44
16 15 12 16 15 12 12 15 13 12	18 21 18 17 15 18 16	21 24 22 19 21	27 23 22 24	26 29 27	33 31	44
13 16 15 13 12 15 15 12 11 13	19 18 17 19 18 15 16	24 21 19 22 19	24 25 22 24	30 26 25	31 33	44
15 13 12 15 15 12 12 13 13 11	17 19 18 17 17 18 15	21 23 23 19 19	25 23 23 23	27 29 25	33 32	44
12 15 15 12 12 13 13 12 12 12	18 17 17 18 18 15 15	23 20 20 21 18	23 26 21 23	29 26 26	31 32	46
15 12 12 13 13 12 12 13 12 11	17 18 18 15 15 18 15	20 23 21 20 18	26 23 21 23	26 29 26	32 31	46
12 13 13 12 12 13 12 11 11 12	18 15 15 18 17 15 13	21 20 20 21 18	23 26 21 21	27 26 26	31 32	46
13 12 12 13 13 11 11 10 10 11	15 18 17 15 14 16 13	20 21 20 19 18	25 23 20 21	25 27 25	31 29	46
12 13 13 11 11 12 10 9 7 10	17 15 15 16 15 13 13	21 19 18 19 17	22 24 19 21	27 25 24	29 31	48
13 11 10 12 10 9 9 10 10 9	15 16 16 14 14 15 13	18 21 20 17 16	23 21 20 21	24 27 23	30 29	48
10 13 10 9 9 10 10 9 9 7	16 14 13 15 15 12 13	20 18 17 19 16	20 23 19 20	26 24 23	29 30	48
10 10 9 10 10 9 9 10 10 6	13 16 15 12 12 15 10	17 19 18 17 16	23 20 18 19	23 25 23	29 28	48
9 10 10 9 9 10 10 9 9 7	15 13 12 15 15 12 10	19 17 17 18 15	20 23 18 19	25 22 23	27 29	50
10 9 9 10 10 9 9 7 7 4	12 15 15 12 12 13 10	17 18 18 17 15	23 20 18 18	22 24 23	29 27	50
9 10 10 9 9 7 7 4 4 7	15 12 12 13 10 10 10	18 17 15 16 15	20 21 17 18	24 22 21	26 28	50
10 9 9 7 7 4 4 7 7 4	12 13 13 11 9 10 10	15 17 16 13 13	19 18 16 17	22 23 19	28 26	50
9 7 7 4 4 7 7 4 4 7	13 11 11 10 10 7 9	16 15 14 15 13	17 20 15 16	22 20 20	25 26	52
7 6 4 7 7 4 4 7 1 4	10 12 10 9 9 10 7	13 16 15 13 12	19 17 15 16	19 22 19	25 24	52
4 7 7 4 4 7 1 4 1 1	10 9 9 10 7 9 7	15 13 12 13 10	16 18 13 15	21 19 18	24 25	52
7 4 4 7 1 4 1 1 1 1	9 10 7 9 4 7 7	12 13 13 11 10	17 15 13 13	18 19 17	24 23	52
4 7 1 4 1 1 1 1 1 1	7 9 6 7 7 4 7	13 11 11 12 9	14 17 12 13	19 17 17	21 23	54
1 4 1 1 1 1 1 1 1 1	6 7 7 6 4 7 4	10 12 10 9 7	15 14 10 12	16 18 15	22 21	54
1 1 1 1 1 1 1 1 -- --	7 4 4 7 1 4 1	10 9 9 7 7	12 13 9 10	16 16 13	20 22	54
1 1 1 1 1 1 1 -- -- --	4 7 1 4 1 1 1	9 7 7 6 7	13 11 7 9	14 15 13	20 19	54
1 1 1 1 1 1 -- -- -- --	1 4 1 1 1 1 1	7 6 6 7 4	10 12 7 7	13 13 12	17 19	56
1 1 1 1 1 -- -- -- -- --	1 1 1 1 1 -- --	4 7 1 4 1	7 9 4 7	12 13 9	18 16	56
1 1 1 1 -- -- -- -- -- --	1 1 1 1 -- -- --	1 4 1 1 1	6 7 1 4	10 10 7	15 16	56
1 1 1 -- -- -- -- -- -- --	1 1 1 -- -- -- --	1 1 1 -- --	1 4 1 1	8 7 4	13 13	56
1 1 -- -- -- -- -- -- -- --	1 1 -- -- -- -- --	1 1 -- -- --	1 1 -- --	1 4 1	10 7	58
1 -- -- -- -- -- -- -- -- --	1 -- -- -- -- -- --	1 -- -- -- --	1 -- -- --	1 -- --	1 6	58
-- -- -- -- -- -- -- -- -- --	-- -- -- -- -- -- --	-- -- -- -- --	-- -- -- --	-- -- --	-- --	55

[3] “pre-sum before Dadda tree” multiplier

```
    4    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -
    5    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -
    8    4    -    -    -    -    -    -    -    -    -    -    -    -    -    -
    8    5    -    -    -    -    -    -    -    -    -    -    -    -    -    -
    8    8    4    -    -    -    -    -    -    -    -    -    -    -    -    -
    9    8    5    -    -    -    -    -    -    -    -    -    -    -    -    -
    9    8    8    4    -    -    -    -    -    -    -    -    -    -    -    -
   10    9    8    5    -    -    -    -    -    -    -    -    -    -    -    -
    5 6  9    8    8    4    -    -    -    -    -    -    -    -    -    -    -
    8   10    9    8    5    -    -    -    -    -    -    -    -    -    -    -
    8    5 6  9    8    8    4    -    -    -    -    -    -    -    -    -    -
    8    8   10    9    8    5    -    -    -    -    -    -    -    -    -    -
    9    8    5 6  9    8    8    4    -    -    -    -    -    -    -    -    -
    9    8    8   10    9    8    5    -    -    -    -    -    -    -    -    -
   10    9    8    5 6  9    8    8    4    -    -    -    -    -    -    -    -
   10    9    8    8   10    9    8    5    -    -    -    -    -    -    -    -
    5 7 10    9    8    5 6  9    8    8    4    -    -    -    -    -    -    -
    8   10    9    8    8   10    9    8    5    -    -    -    -    -    -    -
    8    5 7 10    9    8    5 6  9    8    8    4    -    -    -    -    -    -
    8    8   10    9    8    8   10    9    8    5    -    -    -    -    -    -
    9    8    5 7 10    9    8    5 6  9    8    8    4    -    -    -    -    -
    9    8    8   10    9    8    8   10    9    8    5    -    -    -    -    -
   10    9    8    5 7 10    9    8    5 6  9    8    8    4    -    -    -    -
   10    9    8    8   10    9    8    8   10    9    8    5    -    -    -    -
    5 7 10    9    8    5 7 10    9    8    5 6  9    8    8    4    -    -    -
    8   10    9    8    8   10    9    8    8   10    9    8    5    -    -    -
    8    5 7 10    9    8    5 7 10    9    8    5 6  9    8    8    4    -    -
    8    8   10    9    8    8   10    9    8    8   10    9    8    5    -    -
    9    8    5 7 10    9    8    5 7 10    9    8    5 6  9    8    8    4    -
    9    8    8   10    9    8    8   10    9    8    8   10    9    8    5    -
   10    9    8    5 7 10    9    8    5 7 10    9    8    5 6  9    8    8    4
   10    9    8    8   10    9    8    8   10    9    8    8   10    9    8    5
   11   10    9    8    5 7 10    9    8    5 7 10    9    8    5 6  9    8    8
    6   10    9    8    8   10    9    8    8   10    9    8    8   10    9    8
    -   11   10    9    8    5 7 10    9    8    5 7 10    9    8    5 6  9    8
    -    6   10    9    8    8   10    9    8    8   10    9    8    8   10    9
    -    -   11   10    9    8    5 7 10    9    8    5 7 10    9    8    5 6  9
    -    -    6   10    9    8    8   10    9    8    8   10    9    8    8   10
    -    -    -   11   10    9    8    5 7 10    9    8    5 7 10    9    8    5 6
    -    -    -    6   10    9    8    8   10    9    8    8   10    9    8    8
    -    -    -    -   11   10    9    8    5 7 10    9    8    5 7 10    9    8
    -    -    -    -    6   10    9    8    8   10    9    8    8   10    9    8
    -    -    -    -    -   11   10    9    8    5 7 10    9    8    5 7 10    9
    -    -    -    -    -    6   10    9    8    8   10    9    8    8   10    9
    -    -    -    -    -    -   11   10    9    8    5 7 10    9    8    5 7 10
    -    -    -    -    -    -    6   10    9    8    8   10    9    8    8   10
    -    -    -    -    -    -    -   11   10    9    8    5 7 10    9    8    5 7
    -    -    -    -    -    -    -    6   10    9    8    8   10    9    8    8
    -    -    -    -    -    -    -    -   11   10    9    8    5 7 10    9    8
    -    -    -    -    -    -    -    -    6   10    9    8    8   10    9    8
    -    -    -    -    -    -    -    -    -   11   10    9    8    5 7 10    9
    -    -    -    -    -    -    -    -    -    6   10    9    8    8   10    9
    -    -    -    -    -    -    -    -    -    -   11   10    9    8    5 7 10
    -    -    -    -    -    -    -    -    -    -    6   10    9    8    8   10
    -    -    -    -    -    -    -    -    -    -    -   11   10    9    8    5 7
    -    -    -    -    -    -    -    -    -    -    -    6   10    9    8    8
    -    -    -    -    -    -    -    -    -    -    -    -   11   10    9    8
    -    -    -    -    -    -    -    -    -    -    -    -    6   10    9    8
    -    -    -    -    -    -    -    -    -    -    -    -    -   11   10    9
    -    -    -    -    -    -    -    -    -    -    -    -    -    6   10    9
    -    -    -    -    -    -    -    -    -    -    -    -    -    -   11   10
    -    -    -    -    -    -    -    -    -    -    -    -    -    -    6   10
    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -   11
    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -    6
    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -
    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -
    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -    -
```

```
  4  -  -  -  -  -  -  -  -  -  -  -  -
  5  -  -  -  -  -  -  -  -  -  -  -  -
  8  4  -  -  -  -  -  -  -  -  -  -  -
  8  5  -  -  -  -  -  -  -  -  -  -  -
  8  8  4  -  -  -  -  -  -  -  -  -  -
  9  8  5  -  -  -  -  -  -  -  -  -  -
  9  8  8  4  -  -  -  -  -  -  -  -  -
 10  9  8  5  -  -  -  -  -  -  -  -  -
  5  9  8  8  4  6  -  -  -  -  -  -  -
  8 10  9  8  5  -  -  -  -  -  -  -  -
  8  5  9  8  8  4  6  -  -  -  -  -  -
  8  8 10  9  8  5  -  -  -  -  -  -  -
  9  8  5  9  8  8  4  6  -  -  -  -  -
  9  8  8 10  9  8  5  -  -  -  -  -  -
 10  9  8  5  9  8  8  4  6  -  -  -  -
 10  9  8  8 10  9  8  5  -  -  -  -  -
  5 10  9  8  5  9  8  8  4  7  6  -  -
  8 10  9  8  8 10  9  8  5  -  -  -  -
  8  5 10  9  8  5  9  8  8  4  7  6  -
  8  8 10  9  8  8 10  9  8  5  -  -  -
  9  8  5 10  9  8  5  9  8  8  4  7  6
  9  8  8 10  9  8  8 10  9  8  5  -  -
```

```
  4  -  -  -  -  -  -  -  -
  5  -  -  -  -  -  -  -  -
  8  4  -  -  -  -  -  -  -
  8  5  -  -  -  -  -  -  -
  8  8  4  -  -  -  -  -  -
  9  8  5  -  -  -  -  -  -
  9  8  8  4  -  -  -  -  -
 10  9  8  5  -  -  -  -  -
  5  9  8  8  4  6  -  -  -
  8 10  9  8  5  -  -  -  -
  8  5  9  8  8  4  6  -  -
  8  8 10  9  8  5  -  -  -
  9  8  5  9  8  8  4  6  -
  9  8  8 10  9  8  5  -  -
 10  9  8  5  9  8  8  4  6
 10  9  8  8 10  9  8  5  -
 11  6  7  8  8  8  9  9 10
  8 11  8  8  8  9  9 10 10
 12  9 11  8  8  8  9  9 10
 10 11  9 14  8  9  9 10 10
 14  9 12 11 11  9  9  9 10
 11 12 10 14  9 14  9 10 10
```

```
  4  -  -  -  -  -
  5  -  -  -  -  -
  8  4  -  -  -  -
  8  5  -  -  -  -
  8  8  4  -  -  -
  9  8  5  -  -  -
  9  8  8  4  -  -
 10  9  8  5  -  -
  5  9  8  8  4  6
  8 10  9  8  5  -
  8  6  8  8  8  9
  6 11  8  8  9 10
 11  9 11  8  9  9
  9 12  8 14  9 10
 12 10 14 11 11 10
 10 13 11 14  8 14
 15 11 14 11 13 11
 13 15 12 14 10 14
 15 13 15 12 14 12
 14 17 13 15 12 15
 17 16 15 12 15 12
 16 17 14 17 12 15
```

```
  4  -  -  -
  5  -  -  -
  8  4  -  -
  8  5  -  -
  8  8  4  -
  9  8  5  -
  9  8  8  4
 10  9  8  5
 11  8  8  9
  8 14  9 10
 11 11 14  9
  9 15 11 14
 15 13 15 12
 13 17 13 15
 17 16 16 14
 16 19 14 17
 19 16 17 16
 17 20 16 18
 20 17 18 15
 17 21 17 19
 19 21 19 16
 19 22 18 20
```

```
  8  5  6  7  8  8  8  8  9  9  9 10 10 | 12 10 14 11 14 11 12 10 10 | 18 16 17 16 16 14 | 22 19 22 19
  6 10  9  8  8 10  9  8  8 10  9  8  5 | 10 15 11 14 11 14  9 12 10 | 16 18 16 17 13 17 | 20 23 19 22
  9 11  7  7  8  8  8  8  9  9  9 10 10 | 15 12 14 12 14 11 13 10 11 | 19 17 18 16 17 16 | 23 20 22 19
  7  8 14  8  8  8  8  9  9  9 10 10 10 | 12 15 12 14 11 14 11 14 14 | 17 20 16 18 16 17 | 21 23 20 22
 11 11 11 11  8  8  8  8  9  9  9 10 10 | 16 12 15 12 14 12 14 11 11 | 20 17 18 16 17 16 | 23 22 22 20
  9  9 14  8 14  8  8  9  9  9 10 10 10 | 13 17 13 15 13 15 11 14 14 | 18 20 17 19 14 19 | 22 25 20 23
 11 13 11 11 11 11  8  9  9  9  9 10 10 | 17 16 16 12 15 12 15 11 13 | 21 19 19 17 19 17 | 25 22 23 20
  9 10 14 10 14  8 14  9  9  9 10 10 10 | 14 17 13 17 13 15 12 15 14 | 19 21 18 20 18 19 | 23 25 21 24
 12 14 11 13 11 11 11 11  9  9 10 10 10 | 17 16 17 16 16 12 15 12 14 | 22 19 21 19 19 16 | 25 23 25 22
 10 11 14 10 14 10 14  8 14 10 10 10 10 | 15 17 14 17 13 17 12 16 14 | 19 22 19 20 18 20 | 24 26 23 25
 14 14 11 13 11 12 11 11 11 10 10 10 11 | 17 16 17 16 17 16 16 13 14 | 22 19 22 19 20 19 | 26 24 25 22
 11 11 14 11 14 11 14 10 14 10 10 10 10 | 16 17 15 17 14 17 13 16 14 | 19 22 19 21 19 20 | 24 26 24 25
 12 14 11 13 11 12 11 11 11 10 10 10 11 | 17 16 17 16 17 16 16 13 14 | 22 19 22 19 20 19 | 26 24 25 23
 10 11 12 11 14 10 14 10 14 10 10 10 10 | 15 17 14 17 14 16 13 16 14 | 19 22 19 20 19 20 | 24 26 24 25
 14 13 10 12 11 11 11  9 11 10 10 10 10 | 17 16 16 16 16 14 16 13 14 | 22 19 22 19 20 17 | 26 24 25 22
 11 11 12 11 14 10 14  9  9 10 10 10 10 | 15 17 14 16 13 16 13 15 14 | 19 21 19 20 17 19 | 24 25 24 25
 13 12 10 11 11  9 11  9  9 10 10 10 11 | 17 16 16 14 16 13 15 12 13 | 22 19 20 18 19 18 | 25 23 24 22
 10 10 11 10 14  8  8  9  9  9 10 10 10 | 14 16 13 16 13 15 12 14 14 | 19 20 18 20 17 19 | 24 25 22 24
 11 11  9  8 11  8  9  9  9 10 10 10 11 | 16 13 15 13 15 12 14 11 11 | 20 18 19 18 18 17 | 24 22 24 22
  9  9 11  8  8  8  8  9  9  9 10 10 10 | 13 15 13 15 13 14 12 14 11 | 18 20 17 19 17 18 | 22 24 21 23
  8  7  9  7  8  8  9  9  9 10 10 10 11 | 15 12 15 12 14 12 13 11 11 | 19 17 18 17 17 15 | 23 22 23 21
  6  6 10  9  8  8 10  9  8  8 10  9 -- | 12 13 12 14 12 14 10 12 10 | 17 18 17 18 16 16 | 21 23 20 22
 11 10  9  8  5 10  9  8  5 10  7  7 -- | 13 11 14 11 13 11 12 11 11 | 18 16 17 16 17 15 | 23 20 22 20
  6 10  9  8  8 10  9  8  8 10 -- -- -- | 11 13 12 14 11 14 11 10 10 | 16 18 15 17 15 16 | 20 22 19 21
 11 10  9  8  5 10  9  8  5  7  7 -- -- | 12 11 13 11 11 11 10 10 11 | 17 16 17 16 16 15 | 22 20 22 19
  6 10  9  8  8 10  9  8  8 -- -- -- -- | 10 11 10 14 10  9  9 10 10 | 15 17 14 16 13 15 | 19 21 19 20
 11 10  9  8  5 10  9  8  7 -- -- -- -- | 13  9  8 11  9  9 10 10 11 | 16 16 15 13 15 12 | 21 19 19 18
  6 10  9  8  8 10  9  8 -- -- -- -- -- | 10  6 10  9  8  8 10  9  8 | 15 15 13 14 13 14 | 18 20 18 19
 11 10  9  8  5 10  9  7 -- -- -- -- -- | 11 10  9  8  5 10  9  7 -- | 14 12 15 12 13 12 | 19 17 18 17
  6 10  9  8  8 10  9 -- -- -- -- -- -- |  6 10  9  8  8 10  9 -- -- | 12 15 12 14 10 10 | 17 18 16 17
 11 10  9  8  5 10  7 -- -- -- -- -- -- | 11 10  9  8  5 10  7 -- -- | 13 12 13 11 10 11 | 18 17 17 16
  6 10  9  8  8 10 -- -- -- -- -- -- -- |  6 10  9  8  8 10 -- -- -- | 11 14 10  9 10 10 | 15 17 15 16
 11 10  9  8  5  7 -- -- -- -- -- -- -- | 11 10  9  8  5  7 -- -- -- | 10 11  8  9 10 11 | 16 16 15 13
  6 10  9  8  8 -- -- -- -- -- -- -- -- |  6 10  9  8  8 -- -- -- -- |  8  6 10  9  8  8 | 13 14 13 14
 11 10  9  8 -- -- -- -- -- -- -- -- -- | 11 10  9  8 -- -- -- -- -- | 11 10  9  8 -- -- | 15 12 11 11
  6 10  9  8 -- -- -- -- -- -- -- -- -- |  6 10  9  8 -- -- -- -- -- |  6 10  9  8 -- -- | 12 11  9 10
 11 10  9 -- -- -- -- -- -- -- -- -- -- | 11 10  9 -- -- -- -- -- -- | 11 10  9 -- -- -- | 11  9 10  9
  6 10  9 -- -- -- -- -- -- -- -- -- -- |  6 10  9 -- -- -- -- -- -- |  6 10  9 -- -- -- |  6 10  9 --
 11 10 -- -- -- -- -- -- -- -- -- -- -- | 11 10 -- -- -- -- -- -- -- | 11 10 -- -- -- -- | 11 10 -- --
  6 10 -- -- -- -- -- -- -- -- -- -- -- |  6 10 -- -- -- -- -- -- -- |  6 10 -- -- -- -- |  6 10 -- --
 11 -- -- -- -- -- -- -- -- -- -- -- -- | 11 -- -- -- -- -- -- -- -- | 11 -- -- -- -- -- | 11 -- -- --
  6 -- -- -- -- -- -- -- -- -- -- -- -- |  6 -- -- -- -- -- -- -- -- |  6 -- -- -- -- -- |  6 -- -- --
```

```
  4  -  - |  4  - |  4
  5  -  - |  5  - |  5
  8  4  - |  8  4 | 11
  8  5  - |  8  5 | 14
  8  8  4 | 11  8 | 17
  9  8  5 |  9 14 | 20
 11  8  9 | 15 11 | 21
  9 14 10 | 13 17 | 23
 14 11 11 | 17 16 | 25
 11 15 14 | 16 20 | 27
 17 12 14 | 20 17 | 27
 14 17 15 | 19 21 | 29
 19 16 15 | 22 19 | 29
 17 19 17 | 21 23 | 31
 22 17 17 | 25 21 | 31
 19 22 19 | 24 25 | 32
 22 19 19 | 25 24 | 33
 19 23 20 | 24 26 | 35
 23 20 20 | 26 25 | 35
 20 23 21 | 25 27 | 35
 25 21 21 | 28 25 | 35
 22 25 22 | 27 28 | 37
 25 22 22 | 28 27 | 37
 24 26 23 | 27 30 | 37
 26 24 23 | 30 28 | 38
 24 27 23 | 28 30 | 40
 28 24 23 | 31 29 | 40
 25 28 25 | 30 31 | 40
 28 25 25 | 31 30 | 40
 25 29 25 | 30 32 | 42
 29 26 25 | 32 31 | 42
 27 30 26 | 31 33 | 42
 30 27 26 | 33 32 | 42
 27 30 26 | 32 33 | 44
 30 27 26 | 33 32 | 44
 27 30 26 | 32 33 | 44
 30 27 26 | 33 32 | 44
 27 30 25 | 32 33 | 46
 29 27 25 | 33 32 | 46
 26 30 25 | 31 33 | 46
 28 27 24 | 33 32 | 46
 26 28 24 | 30 32 | 48
 28 25 23 | 31 30 | 48
 25 27 23 | 30 31 | 48
 26 24 23 | 30 29 | 48
 24 26 22 | 28 30 | 50
 26 23 22 | 29 28 | 50
 24 25 21 | 28 30 | 50
 25 22 21 | 28 27 | 50
 22 24 20 | 27 28 | 52
```

23 21 19	27 26	52
20 23 18	25 26	52
23 20 18	26 25	52
20 21 17	25 26	54
21 18 16	24 23	54
18 19 14	23 24	54
17 16 15	22 21	54
14 16 12	19 20	56
15 13 11	19 18	56
12 12 10	17 18	56
11 10 10	16 15	56
6 10 --	13 13	58
11 -- --	11 11	58
6 -- --	6 --	58

REFERENCES

1. L.Dadda,”Some schemes for parallel multipliers”,Alta Frequenza,Volume 34, pp. 346-356.

2. Whitney J.Townsend, Earl E. Swartzlander Jr.,and Jacob A. Abraham,”A comparison of Dadda and Wallace multiplier delays”,Advanced Signal Processing Algorithms,Architectures,and Implementations,Volume 5205, pp. 552-560,Austin,2003.